REMARKS

In the above-identified Office Action, the Examiner rejected Claims 1 - 20 under 35 U.S.C. §102(e) as being anticipated by Oldman.

For the reasons stated more fully below, Applicants submit that the pending claims are allowable over the applied reference. Hence, reconsideration, allowance and passage to issue are respectfully requested.

As disclosed in the SPECIFICATION, with the advent of the Internet and the availability of powerful computers and high speed networks, geographically dispersed computers are beginning to be used as one single unified computing resource. This computing resource is popularly referred to as grid computing (also known as meta-computing, scalable computing, global computing, Internet computing and peer-to-peer computing). In grid computing, resources such as supercomputers, storage systems, data sources etc. which are owned by different organizations are pooled together to solve large-scale computational and data intensive problems in science, engineering and commerce.

To allow for an effective use of grid computing, a grid broker is used. The grid broker performs resource discovery and scheduling of tasks on the distributed grid resources. Thus, if the grid broker receives a request from a computer in New York to process an application program, the grid broker may, based on a resource discovery result, schedule the application program to be processed by a computer in Paris, France.

This concept works fine with application programs that are grid-computing compatible or have been designed to be processed by any computer system on a grid computing system. However, many older application programs (i.e., legacy programs) are monolithic. A monolithic application program is a program that does not rely on outside resources and cannot access or provide services to other applications in a dynamic and cooperative manner. An example of such an application program is a program that may link to specific (local) libraries or read from and/or write to specific remote file systems etc.

Due to the nature of monolithic application programs, they may not run effectively on a grid computing system without modifications. To modify a monolithic program, however, requires that software personnel acquire intimate knowledge of the program. In some instances, this may be quite a time-consuming and work-intensive endeavor, especially, when there is only an object code of the program available and all software personnel who may have designed and developed the program cannot be located.

Consequently, what is needed is a method of executing a monolithic application program successfully on a grid computing system without modifications.

The present invention provides such method. In accordance with the teachings of the invention, in order to execute a monolithic application program on a grid computing system, the monolithic program is first executed on a computer on which the program has previously successfully executed. There, the execution of the program is monitored to collect runtime information of the program. The monitored runtime information is then provided to a computer system on the grid computing system on which the program is to be executed so that the program can be executed on the computer system on the grid computing system successfully.

The invention is set forth in claims of varying scopes of which Claim 1 is illustrative.

> 1. A method of executing a monolithic application program on a grid computing system comprising the steps of:
>
> ***executing the program on a computer on which the program has previously successfully executed***;
>
> ***monitoring the execution of the program to collect runtime information of the program***;
>
> ***providing a computer system on the grid computing system on which the program is to be executed with the runtime information of the program***; and

> ***executing the program on the computer system on the grid computing system***. (Emphasis added.)

The Examiner rejected the claims under 35 U.S.C. §102(e) as being anticipated by Olman. Applicants respectfully disagree.

Oldman purports to teach an adaptive interface for a software program development environment. In accordance with the teachings of Oldman, the software program development environment includes a compiler for producing object code for a given one of a plurality of different platforms from source code for the program. The software program development environment is improved by adding a database that describes the platforms and differences between the platforms. The database is accessible to and interpretable by the compiler and the compiler responds to the source code and the database by indicating whether any of the source code is incompatible with any of the differences described in the database.

In addition to the database, the software program development environment may include a run-time test library that is bound to an application binary produced from the object code and responds to an execution of the application binary by testing whether the application binary is incompatible with any of the differences described in the database. The software program development environment may further include a compatibility run-time library for each of the platforms that is bound to the application binary produced for the platform and that performs conversions necessary for compatibility when the application binary for the platform is executed. Additionally, the software program development environment may include platform proof code for different ones of the platforms, with the platform proof code for a platform being executed on the platform to determine whether the database correctly describes the platform.

However, Oldman does not teach the steps of ***executing the program on a computer on which the program has previously successfully executed***;

monitoring the execution of the program to collect runtime information of the program; ***providing a computer system on the grid computing system on which the program is to be executed with the runtime information of the program***; and ***executing the program on the computer system on the grid computing system*** as claimed.

The Examiner stated that in paragraph [0049], Oldman discloses the step of monitoring the execution of the program. Applicants disagree.

In paragraph [0049], Oldman discloses a compiler 107 for compiling an application source 103 to produce application binary 113, which will execute on a platform 121. But note that the application source code is being compiled and not executed. It is well known in the field that compiling a source program is not equivalent to executing the program.

Thus, Olman does not disclose in that paragraph the step of executing a program much less the step of ***<u>executing</u> the program <u>on a computer on which the program has previously successfully executed</u>***.

The Examiner again used paragraph [0049] to show that Olman teaches the step of monitoring the execution of the program to collect runtime information of the program. Again, Applicants disagree.

As mentioned above, in paragraph [0049] Oldman discloses a compiler 107 for compiling an application source 103 to produce application binary 113, which will execute on a platform 121. Compiling is well known to be quite different than executing a program in the field. Therefore, Olman does not teach the step of ***<u>monitoring the execution of the program</u> to collect runtime information of the program***.

The Examiner then stated that because Olman teaches the use of different computer system platforms on which to compile the application source 103 to produce each time an application binary 113, then Olman teaches a grid of computer system. Applicants once more disagree.

As is well known in the field (see also the Background of the present Invention), a grid of computer systems is a plurality of computer systems that are geographically dispersed and used as one single unified computing resource. In paragraphs [0054] and [0057] as well as in Figs. 2 and 3, a plurality of different computer systems are disclosed and shown. However, the different computer systems are not used as one single unified computing resource. Rather, each computer system is used in the traditional way of using a computer system (i.e., as a standalone computer system).

Therefore, Olman does not teach a grid of computers. much less the step of ***providing a computer system on the grid computing system on which the program is to be executed with the runtime information of the program*** as claimed.

Further, since Olman does not disclose a grid of computer systems, Olman cannot have taught the step of ***executing the program on the computer system on the grid computing system*** as claimed.

Hence, Applicants submit that Claim 1, and its dependent claims are allowable over the applied reference. Independent Claims 6, 11 and 16 as well as their independent claims, which all include the limitations in the above-reproduced Claim 1 are also allowable over the applied reference. Consequently, Applicants once more respectfully request reconsideration, allowance and passage to issue of the claims in the application.

Respectfully Submitted

By: ____________________
Volel Emile
Attorney for Applicants
Registration No. 39,969
(512) 306-7969